

ANALISIS SENTIMEN TERHADAP GAME GENSHIN IMPACT MENGGUNAKAN BERT

¹⁾Ryo Kusnadi, ²⁾Yusuf, ³⁾Andriantony, ⁴⁾Richard Ardian Yaputra, ⁵⁾Melna Caintan

^{1,2,3,4,5)}Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Internasional Batam
^{1,2,3,4,5)}Jl. Gajah Mada, Baloi Permai, Kec. Sekupang, Kota Batam, Kepulauan Riau, Indonesia, 29442
E-mail : ¹⁾ryokusnadi@gmail.com, ²⁾seraphim.laes@gmail.com, ³⁾sesprin6@gmail.com,
⁴⁾richardardian22@gmail.com, ⁵⁾melnacaintan@gmail.com

ABSTRAK

Dengan pesatnya peningkatan jasa internet di jaringan sosial, ada banyaknya informasi dalam jumlah besar terus-menerus dihasilkan secara langsung di saat yang sama. Akhir-akhir ini, analisis sentimen dengan menggunakan ulasan dan pesan telah menjadi topik penelitian yang populer dibicarakan di bidang *Natural Language Processing*. Selama bertahun-tahun, permainan online telah menjadi suatu aktivitas yang tidak bisa dipisahkan dari Sebagian besar orang, terlebih karena gangguan ekonomi yang disebabkan oleh virus Covid-19. Genshin Impact adalah salah satu permainan terkenal yang dikembangkan oleh miHoYo. Penelitian ini berfokus pada analisis sentimen dengan tujuan mengetahui apakah ulasan terpercaya yang dikumpulkan dari *Google Play Store* memiliki sentimen netral, baik atau sentimen buruk sehingga dapat membantu pengembangan permainan kedepannya. Diperlukan proses klasifikasi analisis sentimen otomatis untuk mengurangi kesalahan yang disebabkan oleh sumber daya manusia. Meskipun demikian, sangat jarang ditemukan studi yang membahas *feature extraction* dan *deep learning models* yang sesuai dengan kasus ini, terutama dalam bisnis permainan. Tahap proses penelitian ini adalah pengekstraksian data melalui *Google Play Store*, dan menggunakan *Bidirectional Encoder Representations from Transformers* (BERT) sebagai model kecerdasan buatan.

Kata Kunci: Analisis Sentimen, Klasifikasi, BERT, Ilmu Data

ABSTRACT

By huge improvement of Internet Services on social networking, there are a lot data that were streamly made in every time. Recently, sentiment analysis by using online reviews and messages has become a popular research issue in Natural Language Processing field. Over the years, online game have become inseparable thing for most of us, especially in the widespread economic disruption caused by the Covid-19. Genshin Impact is one of the well-known game that developed by miHoYo. This research focused on sentiment analysis with the purpose to find out whether the respected review that scraps from google play store has a neutral, positive or negative sentiment so it will be helpful for afterward game improvement. An autonomus sentiment analysis classification process is required to reduce human error. But, it is hard to get studies that explore about the extraction features and the deep learning models that fit with this case, especially in the business game. This research process stage is data scrapping through the google play store, and using Bidirectional Encoder Representations from Transformers (BERT) as the machine learning model.

Keyword: Sentiment Analysis, Classification, BERT, Data Science

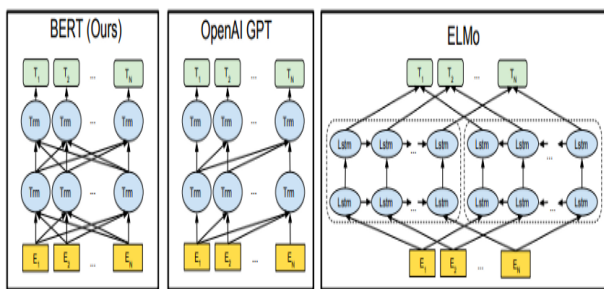
PENDAHULUAN

Selama bertahun-tahun, game online menjadi hal yang tidak dapat dipisahkan bagi sebagian besar dari kita, terutama di tengah meluasnya gangguan ekonomi akibat virus Covid-19. Dengan praktik social distancing yang mengurangi aktivitas konsumen dan bisnis seminimal mungkin, game menawarkan aktifitas yang menarik bagi orang-orang di rumah yang mencari interaksi sosial, dan riset menunjukkan bahwa terjadinya pertumbuhan besar dalam waktu bermain game dan penjualan sejak lockdown dimulai [1]. Saat ini,

Orang biasanya memeriksa ulasan dan peringkat game sebelum bermain di perangkat mereka [2]. Analisis sentimen sangat berguna saat pihak pengembang game ingin mengetahui opini pelanggan tentang pengalaman bermain game. Jadi dengan membaca ulasan dari platform adalah cara paling cocok untuk memutuskan arah pengembangan dan peningkatan game selanjutnya. Analisis sentimen juga dapat memverifikasi popularitas produk [3]. *Google Play* adalah salah satu platform tempat pengguna dapat meninjau pengalaman bermain game mereka. Analisis

sentimen dapat digunakan untuk mengetahui pendapat mereka tentang game tersebut apakah sentimen positif, netral, dan negatif [4]. Dengan kata lain, kita dapat menggunakan analisis sentimen untuk menganalisis dan menghasilkan dokumen terperinci yang merangkum pendapat masukan. Dengan memiliki data tersebut, pengembang game dapat memperoleh informasi tersebut sebagai suatu keuntungan [5].

Kebanyakan frasa ulasan di *Google Play Store* adalah ketidakkonsistenan, mengandung banyak kekacauan, dan redundansi. Sulit untuk melakukan data preprocessing sebelum melakukan analisis sentimen [6]. Oleh karena itu data preprocessing merupakan suatu proses yang biasanya membutuhkan banyak sumber daya dan menghabiskan banyak waktu untuk itu. Pada tahap preprocessing data juga terdapat transformasi data yang meliputi generalisasi data, *smoothing*, normalisasi dan konstruksi atribut. Dan juga perlu dilakukan penanganan *imbalanced data*, karena *imbalanced class* akan menyebabkan akurasi menjadi tidak akurat [7]. Akurasi algoritma dapat ditingkatkan setelah data preprocessing [8].



Gambar 1. Arsitektur BERT dibandingkan dengan Arsitektur model lainnya

Penelitian sebelumnya menggunakan *Naïve Bayes Classifier* (NBC) sebagai teknik machine learning untuk analisis sentimen. Input data untuk penelitian tersebut berasal dari komentar aplikasi di *Google Play Store* [9]. *Data Training* dihitung probabilitas

kemunculan setiap kata di setiap kelas yang telah ditentukan. Baru-baru ini, model bahasa pra-trlatih seperti *Bidirectional Encoder Representations from Transformers* (BERT) [10], *Universal Language Model Fine-tuning* (ULMFiT) [11], dan *Embeddings from Language Models* (ELMo) [12] telah menunjukkan akurasi yang tinggi dalam ekspresi semantik teks. Di antara model tersebut, BERT telah mencapai hasil yang sangat baik dalam *sentence-level sentiment classification* (SST-2) [13]. Dalam tulisan ini, penulis akan menggunakan BERT untuk mengganti *word embeddings* dengan representasi kontekstual yang sesuai. BERT sangat mampu menangkap banyak koleksi informasi sintaksis lintas bahasa [14]. Secara khusus, kami menghasilkan *embedding* BERT dari token WordPiece pertama sebagai *word embedding*, dan melatih parser BiAffine dalam konfigurasi defaultnya. Sehingga dengan penerapan model BERT, dapat membantu para pengembang game seperti *Genshin Impact* dan pengembang lainnya untuk memanfaatkan model ini sebagai referensi data dan kemajuan dalam melakukan penentuan keputusan dalam pengembangan game.

METODE

Setiap *Data Scientist* Membutuhkan cara atau metodologi untuk menyelesaikan sebuah masalah Data. Dibutuhkan sebuah sistem iteratif dari sebuah metode yang memandu *data scientist* tentang pendekatan ideal agar seorang *Data Scientist* dapat memecahkan masalah dengan menggunakan ilmu data, Dalam penelitian ini, penulis menggunakan metode *Cross-Industry Standard Process for Data Mining* (CRISP-DM). Terdapat beberapa tahapan proses yang perlu dilewati agar penelitian ini dapat berjalan dengan baik [15], antara lain:

Business Understanding

Pada tahap ini penulis memulai dengan pemahaman bisnis. Dengan kata lain, penulis perlu menetapkan tujuan, sasaran dan membuat daftar kebutuhan bisnis yang menunjukkan urgensi permintaan ini. Setelah setiap kebutuhan bisnis didefinisikan dan ditentukan, kita perlu melakukan analisis dan mengidentifikasi teknik yang cocok untuk mencapai hasil yang diinginkan. Tujuan dari penelitian ini adalah Analisis informasi subjektif yang terkandung dalam teks serta Memberikan wasasan terkait pendapat dan emosi tentang situasi game terkini.

Data Understanding

Pada tahap ini penulis akan menentukan data-data apa saja yang perlu penulis gunakan. Dimulai dari isi, format dan juga sumber. karena data yang kita miliki akan berpengaruh pada algoritma yang digunakan [16]. Setelah kebutuhan data ditentukan, penulis akan melakukan scrapping data dari layanan distribusi digital berbagai jenis aplikasi game android yang bernama *Google Play* menggunakan *package google play scrapper*. Hal yang perlu dilakukan oleh penulis yaitu menginisialisasi nama package aplikasi untuk kasus ini yaitu *com.miHoYo.GenshinImpact*. disini kita hanya perlu melakukan *tqdm* untuk melakukan *scrapping data*, dan kita harus melakukan pengurutan berdasarkan *score* dan *review* terbaru agar dataset yang akan kita punyai terstruktur dengan baik. Dikarenakan topik ini merupakan *sentiment analysis*, maka data utama yang diperlukan hanya *review* dan *score*. Pada saat melakukan *scrapping data*, penulis juga mengambil data tersebut dengan persebaran merata sesuai dengan nilai *review*.

Data Preparation

Pada tahap ini, penulis perlu mempersiapkan data kita dari tahap masuk ke

pemodelan. Dimulai dari Tambahkan token khusus untuk memisahkan kalimat dan memberi tahu model bahwa kita melakukan klasifikasi. Kita memanfaatkan *hugging face* agar bisa memenuhi kebutuhan tersebut. Setelah itu kita perlu membuat *array 0s (pad token)* dan *1s (real token)* yang disebut "*attention mask*" dan mengubah dataset tersebut dalam bentuk *Pytorch*, lalu mengsplit data yang ada menjadi dua dataset (*data training* dan *test*).

Modeling

Ditahap ini penulis akan melakukan modelling menggunakan model *Bidirectional Encoder Representations from Transformers* (BERT), jika performa model tersebut tidak sesuai harapan, maka penulis akan melakukan *Data preparation*. Disini penulis menggunakan fungsi softmax karena hasil akhir dari prediksi ini merupakan vektor yang mewakili distribusi probabilitas dari daftar hasil potensial.

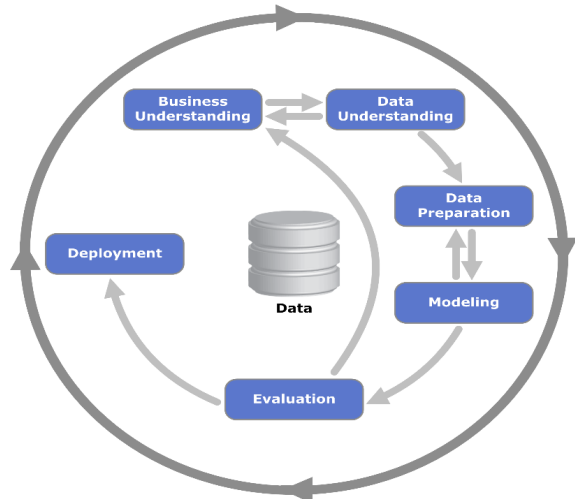
Evaluation

Pada tahap ini penulis akan mengevaluasi performa model yang telah dibuat. Jika performa model kurang memuaskan, maka akan kembali lagi ke dalam tahap *Data Preparation*.

Deployment

Pada tahap ini penulis akan melakukan uji coba untuk membuktikan kebenaran ataupun akurasi dari model ini.

Flowchart tahapan dalam penelitian ini dapat dilihat digambar dibawah berikut:



Gambar 2. Flowchart Tahapan Penelitian

HASIL

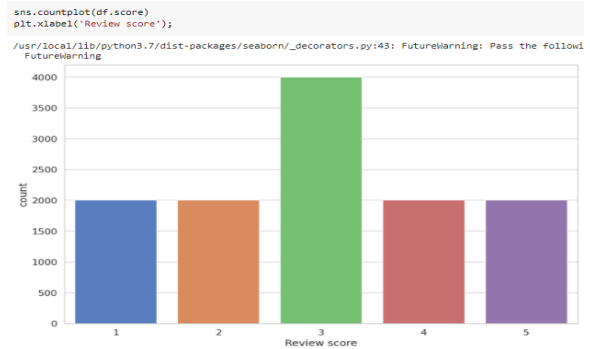
Data Scrapping

Penulis menggunakan *google-play-scraper Package* yang tersedia dalam Python dan menggunakan *Google Collab* yang didalamnya tersedia *Jupyter Notebook* sebagai notebook dalam projek ini. Di dunia yang ideal, kita akan mendapatkan banyak ulasan positif, netral dan negatif. Tapi dengan adanya banyaknya data tersebut, tidaklah efisien untuk menggunakan semua data [17]. Maka penulis memutuskan untuk mempunyai data yang seimbang dengan menggunakan opsi *package* scraping untuk memfilter skor ulasan. Berikut adalah distribusi data yang discrapping dari *Google Play*:

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12000 entries, 0 to 11999
Data columns (total 12 columns):
#   Column                Non-Null Count  Dtype
---  ---                ---
0   reviewId              12000 non-null  object
1   userName              12000 non-null  object
2   userImage             12000 non-null  object
3   content               12000 non-null  object
4   score                 12000 non-null  int64
5   thumbsUpCount         12000 non-null  int64
6   reviewCreatedVersion  10524 non-null  object
7   at                    12000 non-null  object
8   replyContent          1 non-null      object
9   repliedAt             1 non-null      object
10  sortOrder              12000 non-null  object
11  appId                 12000 non-null  object
dtypes: int64(2), object(10)
memory usage: 1.1+ MB
    
```

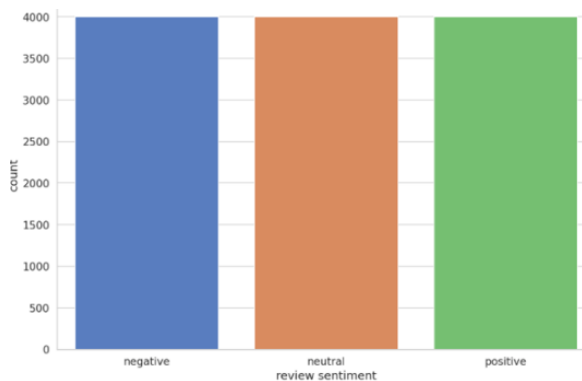
Gambar 3. Informasi Dataset Scrapping Genshin Impact



Gambar 4. Persebaran Distribusi Dataset Scrapping Genshin Impact

Data Preparation

Ditahap ini penulis akan menghilangkan nilai null dan duplikat. dan juga di tahap ini kita akan melakukan *class normalization* untuk menghindari *class imbalance* dimana review sebelumnya terdiri review rating sebelumnya terdiri dari nilai 1 hingga 5, mengklasifikasikannya menjadi negatif(1-2),netral(3), dan positif(4-5). Hal lain yang akan dilakukan oleh yaitu Tambahkan token khusus untuk memisahkan kalimat. Token [SEP] untuk menandakan akhir dari sebuah kalimat, Token [CLS] digunakan untuk memberitahu model bahwa kita sedang melakukan klasifikasi dan penanda dari awal kalimat. Token [PAD] untuk *padding*, token [UNK] untuk data yang tidak diketahui. Kita bisa menggunakan semua fungsi tersebut menggunakan function `tokenizer.encode_plus()`. Setelah itu kita perlu menginisialisasi suatu urutan karena model BERT bekerja dengan panjang *sequence* yang statik. Setelah itu, kita melakukan split data. Untuk menggunakan model BERT, kita menggunakan *Helper* dari *Transformer*. Disini kita perlu perhatikan array 0s (*pad token*) dan 1s (*real token*) yang disebut *attention mask* [18] memiliki panjang nilai yang sama dengan nilai input.



Gambar 5. Persebaran Distribusi Dataset Setelah normalisasi kelas

Modelling Menggunakan BERT

Ada banyak *helper* yang menjadikan BERT mudah digunakan dengan *library* Transformers. Contohnya seperti *BertForSequenceClassification*, *BertForQuestionAnswering* ataupun yang lainnya. tetapi untuk model ini penulis akan menggunakan Model BERT dasar dan membuat klasifikasi sentimen di atasnya. Klasifikasi model akan mendelegasikan sebagian besar tugas berat ke BertModel. Kami menggunakan *dropout layer* untuk beberapa *regularization* dan lapisan yang sepenuhnya terhubung untuk output model. Kita menggunakan *regularization* untuk melakukan modifikasi pada model neural network yang bertujuan untuk mengurangi generalization error. Dan kami menyimpan tersebut dalam *dropout layer*. dimana *dropout layer* merupakan *layer* yang digunakan untuk mencegah terjadinya *overfitting* dan juga mempercepat proses learning. Dapat diperhatikan juga *raw output* dari layer terakhir dibutuhkan untuk *cross-entropy loss function* agar pytorch dapat berjalan dengan baik. *cross-entropy loss function* adalah fungsi loss default yang digunakan untuk masalah klasifikasi biner.

Untuk mendapatkan probabilitas prediksi dari model yang telah ditraining, kita akan mengaplikasikan fungsi softmax kedalam

output [19]. Untuk mereproduksi training procedure penulis akan menggunakan AdamW Optimizer yang disediakan oleh Hugging Face. Penerapan ini akan mengoreksi *weight decay* [20], dan untuk hyperparameternya *fine-tuning* yaitu sebagai berikut kami menggunakan sejumlah 32 *Batch size*, *Learning rate (Adam): 2e-5* dan Jumlah *epochs* yang dipakai: 10.

```
F.softmax(model(input_ids, attention_mask), dim=-1)
tensor([[0.4058, 0.1836, 0.4106],
        [0.5049, 0.1409, 0.3542],
        [0.2603, 0.1502, 0.5895],
        [0.5575, 0.1270, 0.3156],
        [0.5735, 0.1864, 0.2402],
        [0.3538, 0.1587, 0.4875],
        [0.3799, 0.1083, 0.5117],
        [0.3146, 0.1831, 0.5023],
        [0.3397, 0.1613, 0.4990],
        [0.2510, 0.2329, 0.5161],
        [0.4378, 0.1679, 0.4143],
        [0.4436, 0.1989, 0.3575],
        [0.5657, 0.0915, 0.3428],
        [0.4811, 0.1773, 0.3416],
        [0.4003, 0.1804, 0.4193],
        [0.3722, 0.2221, 0.4057]])
```

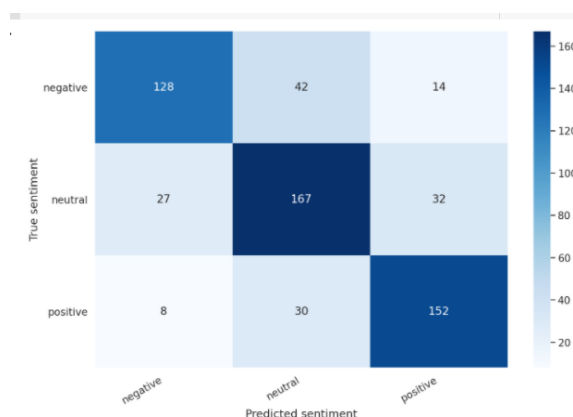
Gambar 6. Implementasi Softmax kedalam Model

Evaluation

Berikut adalah *classification report* dari model yang sebelumnya dibuat.

	precision	recall	f1-score	support
negative	0.79	0.70	0.74	184
neutral	0.70	0.74	0.72	226
positive	0.77	0.80	0.78	190
accuracy			0.74	600
macro avg	0.75	0.74	0.75	600
weighted avg	0.75	0.74	0.74	600

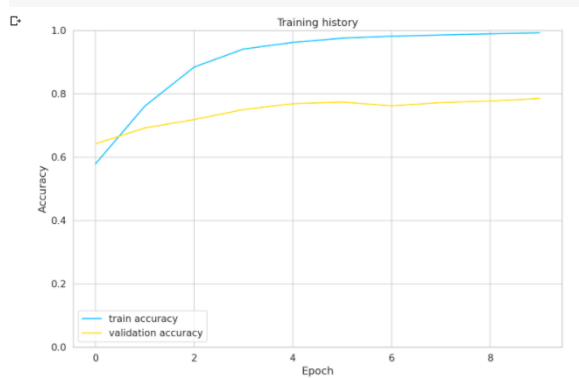
Gambar 7. Classification Report



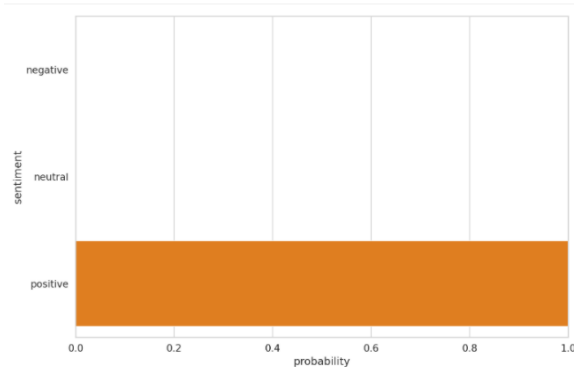
Gambar 8. Confusion Matrix

Dari *Classification Report*, model yang

dibuat memiliki presisi sebesar 0.79 untuk negatif, positif 0.77 dan 0.70 untuk netral. Nilai netral susah diprediksi karena berada di ambang pertengahan antara positif dan negatif. Sedangkan untuk nilai *support* memiliki nilai persebaran yang lumayan merata, yaitu 184 untuk negatif, 190 untuk positif dan 226 untuk netral. Dari nilai *support* kita dapat menyimpulkan tidak adanya *class imbalance* dalam model ini. Untuk *Confusion matrix* diatas, kita dapat menyimpulkan bahwa untuk mengklasifikasikan ulasan netral lebih sulit dibandingkan nilai *sentiment* lainnya.



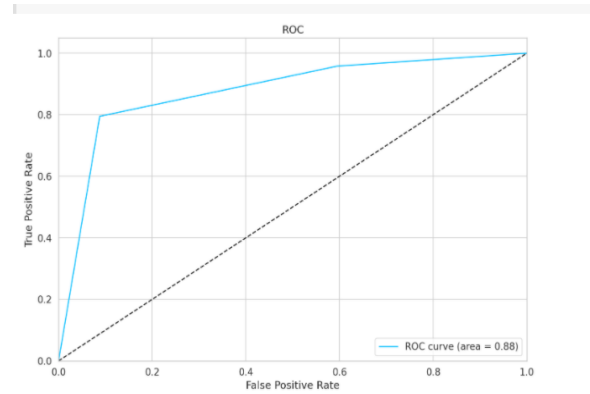
Gambar 9 Training History



Gambar 10. Unit Test

Untuk *training history* model BERT, model ini ditraining sebanyak 10 *epoch* sehingga akurasi memiliki nilai yang cukup baik yaitu sehingga model ini memiliki peningkatan akurasi yang berbanding lurus dengan jumlah *epoch* yang ditraining seperti yang tertera di gambar 3.5. Di gambar 3.6 kami melakukan testing menggunakan beberapa

kata-kata positif sehingga memiliki nilai probabilitas positif seperti yang tertera digambar.



Gambar 11. ROC

Dari gambar ROC (*Receiver Operating Characteristics*) diatas, nilai yang didapat untuk ROC curve yaitu 0.88. dapat disimpulkan bahwa *performance* untuk *classification problem* dalam menentukan *threshold* dari model BERT dalam kasus ini cukup baik.

Deployment

Dari hasil modeling yang dilakukan sebelumnya dengan menggunakan BERT, kita dapat menggunakan model tersebut untuk dalam penentuan keputusan apakah *review* yang ditulis oleh *user* sehingga tidak terjadinya perbedaan keputusan antara suatu *reviewer* dengan *reviewer* lainnya. hal ini dapat berguna untuk perusahaan game Genshin Impact dalam pengembangan game kedepannya.

Review text: I intrested about this app
Sentiment : positive

Gambar 12. Contoh Prediction

KESIMPULAN

Berdasarkan pengujian dan analisis penggunaan *sentiment analysis* Terhadap Game Genshin Impact menggunakan model *Bidirectional Encoder Representations from Transformers* (BERT) maka kita dapat

menyimpulkan :

1. Dengan menggunakan model BERT penulis berhasil merancang dan membangun prediksi sentimen baik itu positif, negatif ataupun netral untuk memprediksi sebuah kalimat yang berasal dari *user*.
2. BERT adalah *Pre-train Natural Language Model* yang berarti seperti membuat kotak hitam yang memahami suatu bahasa kemudian model tersebut diminta untuk melakukan suatu tugas spesifik tertentu. BERT dapat mengungguli 11 tugas NLP yang paling umum setelah *fine-tuning*, yang pada dasarnya menjadi pendorong untuk *Natural Language Processing and Understanding*.
3. Penelitian ini menggunakan data yang berasal dari *google playstore* dengan game yang berjudul Genshin Impact. Dengan data tersebut penulis dapat membuat model analisis sentimen yang berasal dari sebuah kata.
4. Perhitungan performa model analisis sentimen dengan metode BERT dari 12000 dataset. Dataset tersebut dibagi menjadi perhitungan klasifikasi kategori yang terdiri atas positif negatif dan netral, dan juga perhitungan klasifikasi sentimen. Untuk hasil dari model, model lebih mudah mendeteksi nilai positif karena memiliki nilai presisi tertinggi yaitu sebanyak 0.86%, 0.78% recall dan 0.82% di *f1-score* sehingga model tersebut mampu mengembalikan nilai sentimen dengan baik. Akan tetapi dapat dilihat dari data akurasi, nilai netral lebih susah dideteksi karena memiliki nilai akurasi yang lebih rendah dibandingkan kategori yang lain.

DAFTAR PUSTAKA

- [1] D. L. King, P. H. Delfabbro, J. Billieux, and M. N. Potenza, "Problematic Online Gaming and The COVID-19 Pandemic," *J. Behav. Addict.*, vol. 9, no. 2, 2020.
- [2] S. Chakraborty, I. Mobin, A. Roy, and M. H. Khan, "Rating Generation of Video Games using Sentiment Analysis and Contextual Polarity from Microblog," *Proc. Int. Conf. Comput. Tech. Electron. Mech. Syst. CTEMS 2018*, pp. 157–161, 2018, doi: 10.1109/CTEMS.2018.8769149.
- [3] L. Yang, Y. Li, J. Wang, and R. S. Sherratt, "Sentiment Analysis for E-Commerce Product Reviews in Chinese Based on Sentiment Lexicon and Deep Learning," *IEEE Access*, vol. 8, pp. 23522–23530, 2020, doi: 10.1109/ACCESS.2020.2969854.
- [4] Y. Wang, K. T. Kim, B. J. Lee, and H. Y. Youn, "Word clustering based on POS feature for efficient twitter sentiment analysis," *Human-centric Comput. Inf. Sci.*, vol. 8, no. 1, 2018, doi: 10.1186/s13673-018-0140-y.
- [5] D. Blazquez and J. Domenech, "Big Data sources and methods for social and economic analyses," *Technol. Forecast. Soc. Change*, vol. 130, no. March, pp. 99–113, 2018, doi: 10.1016/j.techfore.2017.07.027.
- [6] S. Pradha, M. N. Halgamuge, and N. Tran Quoc Vinh, "Effective text data preprocessing technique for sentiment analysis in social media data," *Proc. 2019 11th Int. Conf. Knowl. Syst. Eng. KSE 2019*, pp. 1–8, 2019, doi: 10.1109/KSE.2019.8919368.
- [7] K. Fithriasari, I. Hariastuti, and K. S. Wening, "Handling Imbalance Data in Classification Model with Nominal Predictors," *Int. J. Comput. Sci. Appl. Math.*, vol. 6, no. 1, p. 33, 2020, doi: 10.12962/j24775401.v6i1.6643.
- [8] Z. Jianqiang and G. Xiaolin, "Comparison research on text preprocessing methods on twitter sentiment analysis," *IEEE Access*, vol. 5, no. c, pp.

- 2870–2879, 2017, doi: 10.1109/ACCESS.2017.2672677.
- [9] S. Wahyu Handani, D. Intan Surya Saputra, Hasirun, R. Mega Arino, and G. Fiza Asyrofi Ramadhan, “Sentiment analysis for go-jek on google play store,” *J. Phys. Conf. Ser.*, vol. 1196, no. 1, 2019, doi: 10.1088/1742-6596/1196/1/012032.
- [10] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” *NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.*, vol. 1, no. Mlm, pp. 4171–4186, 2019.
- [11] J. Howard and S. Ruder, “Universal language model fine-tuning for text classification,” *ACL 2018 - 56th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf. (Long Pap.)*, vol. 1, pp. 328–339, 2018, doi: 10.18653/v1/p18-1031.
- [12] M. E. Peters *et al.*, “Deep contextualized word representations,” *NAACL HLT 2018 - 2018 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.*, vol. 1, pp. 2227–2237, 2018, doi: 10.18653/v1/n18-1202.
- [13] Z. Gao, A. Feng, X. Song, and X. Wu, “Target-dependent sentiment classification with BERT,” *IEEE Access*, vol. 7, pp. 154290–154299, 2019, doi: 10.1109/ACCESS.2019.2946594.
- [14] D. Kondratyuk and M. Straka, “75 Languages, 1 Model: Parsing Universal Dependencies Universally,” *arXiv*, pp. 2779–2795, 2019.
- [15] E. S. Palupi and S. M. Pahlevi, “Klasifikasi Opportunity Menggunakan Algoritma C4.5, C4.5 dan Naive Bayes Berbasis Particle Swarm Optimization,” *Inti Nusa Mandiri*, vol. 14, no. 2, pp. 133–138, 2020.
- [16] Ainurrohmah, “Akurasi Algoritma Klasifikasi pada Software Rapidminer dan Weka,” vol. 4, pp. 493–499, 2021.
- [17] E. Sutoyo and M. A. Fadlurrahman, “Penerapan SMOTE untuk Mengatasi Imbalance Class dalam Klasifikasi Television Advertisement Performance Rating Menggunakan Artificial Neural Network,” *JEPIN (Jurnal Edukasi dan Penelit. Inform.)*, vol. 6, no. 3, pp. 379–385, 2020.
- [18] Y. Guan, J. Leng, C. Li, Q. Chen, and M. Guo, “How Far Does BERT Look At: Distance-based Clustering and Analysis of BERT’s Attention,” pp. 3853–3860, 2021, doi: 10.18653/v1/2020.coling-main.342.
- [19] R. A. Pangestu, B. Rahmat, and F. T. Anggraeny, “Implementasi Algoritma CNN untuk Klasifikasi Citra Lahan dan Perhitungan Luas,” *Inform. dan Sist. Inf.*, vol. 1, no. 1, pp. 166–174, 2020.
- [20] X. Li, S. Chen, Y. Xia, and J. Yang, “Understanding the disharmony between weight normalization family and weight decay: ϵ -shifted L2 regularizer,” *arXiv*, vol. 1, 2019.